💡 React Context API + useContext Hook

📌 Purpose:

> Component tree me deeply nested children ko data dena without prop drilling

---

🧱 Breakdown of Steps

🧾 Step 1: Create Context

```
const UserContext = createContext();
```

> This creates a context object that you can provide and consume

---

🧾 Step 2: Wrap Component Tree with Provider

```
<UserContext.Provider value={user}>
  <ChildA />
</UserContext.Provider>
```

> Value ke form me data pass karte ho — jo sab children access kar sakte hain

---

🧾 Step 3: Provide Some Value (State / Object / String / Anything)

```
const [user, setUser] = useState({
  name: "priyansh"
});
```

> Ye value UserContext.Provider me diya jaata hai

🧾 Step 4: Consume with `useContext` (No prop needed!)

```
const user = useContext(UserContext);
```

> Bas ek line aur directly context ka data mil gaya — bina prop pass kare

---

🧩 Component Structure:

```
App (Provider)
  └── ChildA
        └── ChildB
              └── ChildC (🎯 Data consumed here)
```

🧑‍💻 Code Flow:

📁 `App.jsx`

```
import { createContext, useState } from 'react';
import ChildA from './components/ChildA';

const UserContext = createContext();

function App() {
  const [user, setUser] = useState({ name: "priyansh" });
  return (
    <UserContext.Provider value={user}>
      <ChildA />
    </UserContext.Provider>
  );
}

export default App;
export { UserContext };
```

📁 `ChildA.jsx`

```
import ChildB from './ChildB';

const ChildA = () => {
  return <ChildB />;
};

export default ChildA;
```

📁 `ChildB.jsx`

```
import ChildC from './ChildC';

const ChildB = () => {
  return <ChildC />;
};

export default ChildB;
```

📁 `ChildC.jsx`

```
import { useContext } from 'react';
import { UserContext } from '../App';

const ChildC = () ⇒ {
  const user = useContext(UserContext);

  return (
    <div>I love you {user.name} with love from Nafeesa</div>
  );
};

export default ChildC;
```

🔥 Advantages of useContext:

✅ Avoids **prop drilling**

✅ Makes data **easily sharable** across component tree

✅ Clean and readable code

🧠 Real Life Example:

> Socho App me theme, auth, ya language — sab bar bar pass nahi karna
>
> Ek context banao aur har jagah use karo.

✅ Summary Box:

Concept	Usage
`createContext()`	To create the context
`Context.Provider`	To wrap children & provide value

`useContext(Context)`	To consume context value directly